

Extension and Exchange Schemas

Modules Roadmap: You Are Here

NIEM Overview

IEPD Concepts

How NIEM uses XML (pt. 1)

How NIEM uses XML (pt. 2)

Business Skills

Exchange Content Modeling

Mapping

Subsets

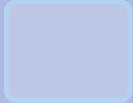
 **Extension and Exchange Schemas**

Packaging and Distribution

Implementation Considerations

Objectives Roadmap

This module supports the following course objective:



Describe what NIEM is.



Describe what an IEPD is.



Comprehend artifacts included in an IEPD.



Develop artifacts included in an IEPD.



Package an IEPD.



Understand advanced XML concepts, as required by NIEM.



Recognize business skills required to successfully participate in an IEPD development project.

Module Objectives

- After completing this module, you should be able to:
 - ◆ Define extension schemas.
 - ◆ Explain the role of extension schemas.
 - ◆ Develop an extension schema.
 - ◆ Differentiate which techniques to use and when creating extension schemas.

What is an Extension Schema?

- Extension schemas define elements you need that are not in NIEM.
- Technically an optional artifact but in practice is nearly always needed.

Creating an Extension Schema

- What the extension schema needs:
 - ◆ Its own namespace.
 - ◆ References to NIEM and W3C namespaces.
 - ◆ An import statement to import NIEM namespaces.
 - ◆ NIEM Conformance statement.

Extension Schema Shell

```
<xsd:schema targetNamespace="http://mynamespace.com/extension/1.0"
  xmlns:local-ns="http://mynamespace.com/extension/1.0"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:nc="http://niem.gov/niem/niem-core/2.0" xmlns:i="http://niem.gov/niem/appinfo/2.0"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:annotation>
    <xsd:documentation>Sample extension schema</xsd:documentation>
    <xsd:appinfo>
      <i:ConformantIndicator>>true</i:ConformantIndicator>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:import namespace="http://niem.gov/niem/niem-core/2.0"
  schemaLocation="niem/niem-core/2.0/niem-core.xsd"/>

  <!-- Define Types Here -->

  <!-- Define Elements Here -->

</xsd:schema>
```

How do you make one?

```
<xsd:schema> <!-- namespace references defined here -->
  <xsd:import namespace="http://niem.gov/niem/niem-core/2.0"
    schemaLocation="niem/niem-core/2.0/niem-core.xsd"/>

  <xsd:complexType name="HybridVehicleType">
    <xsd:complexContent>
      <xsd:extension base="nc:VehicleType"/>
      <xsd:sequence>
        <xsd:element ref="ext:FuelCellVoltageText"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent >
</xsd:complexType>

<xsd:element name="HybridVehicle" type="ext:HybridVehicleType" />
<xsd:element name="FuelCellVoltageText" type="nc:TextType" />

</xsd:schema>
```

Exercise 16.1: Extension Schema

- Create an extension schema to extend `nc:PersonType` and add a property for Sense of Humor.

Solution 16.1: Extension Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://ijis.org/niem2/example/1.0"
  xmlns:local-ns="http://ijis.org/niem2/example/1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:nc="http://niem.gov/niem/niem-core/2.0">
  <xs:import schemaLocation="niem/niem-core/2.0/niem-core.xsd"
    namespace="http://niem.gov/niem/niem-core/2.0"/>

  <xs:complexType name="PersonType">
    <xs:complexContent>
      <xs:extension base="nc:PersonType">
        <xs:sequence>
          <xs:element ref="local-ns:SenseOfHumorText"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:element substitutionGroup="nc:Person" name="Person" type="local-ns:PersonType"/>
  <xs:element name="SenseOfHumorText" type="nc:TextType"/>

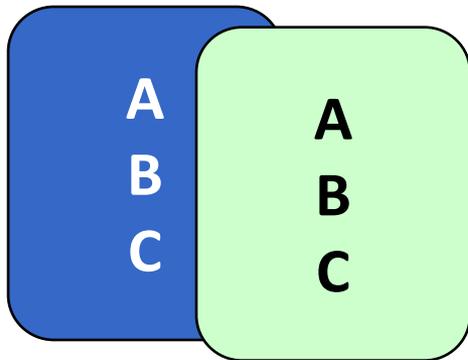
</xs:schema>
```

Architecture Decisions

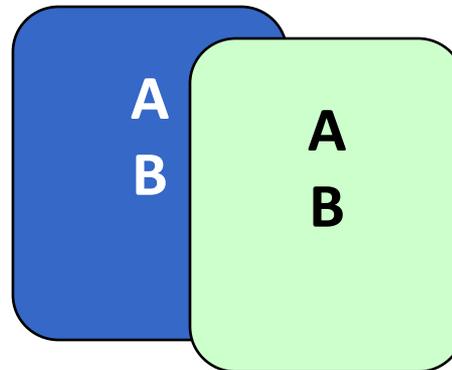
- Exchange-Specific Schemas
 - ◆ Each exchange has its own extension schema.
- Agency-Wide Schema
 - ◆ The agency has one large extension schema (or set of schemas) containing all extensions.
 - ◆ Sometimes used in conjunction with an exchange specific schema.
- Multiple Extension Schemas

Exchange Specific Schema

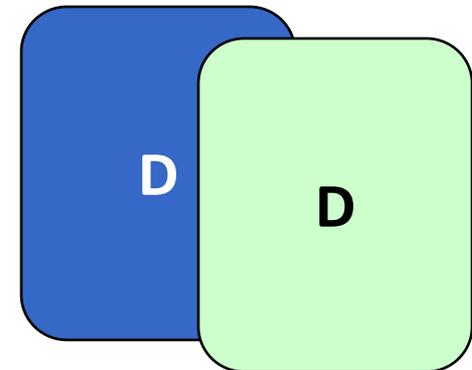
- Provides more granular validation for a particular exchange.
- Does not maximize re-use.



Exchange 1



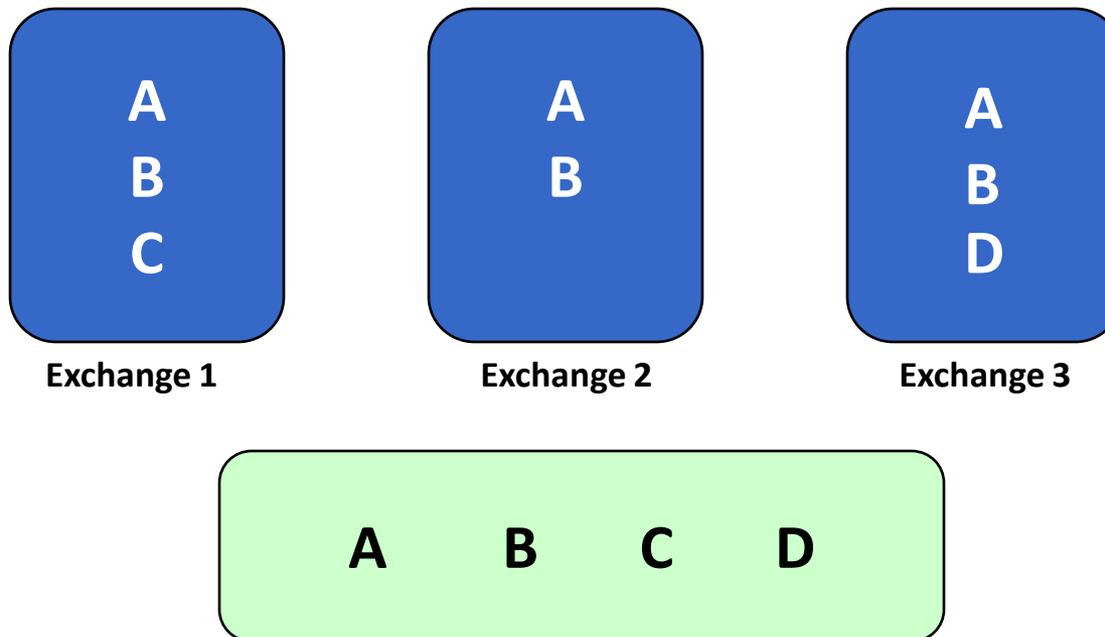
Exchange 2



Exchange 3

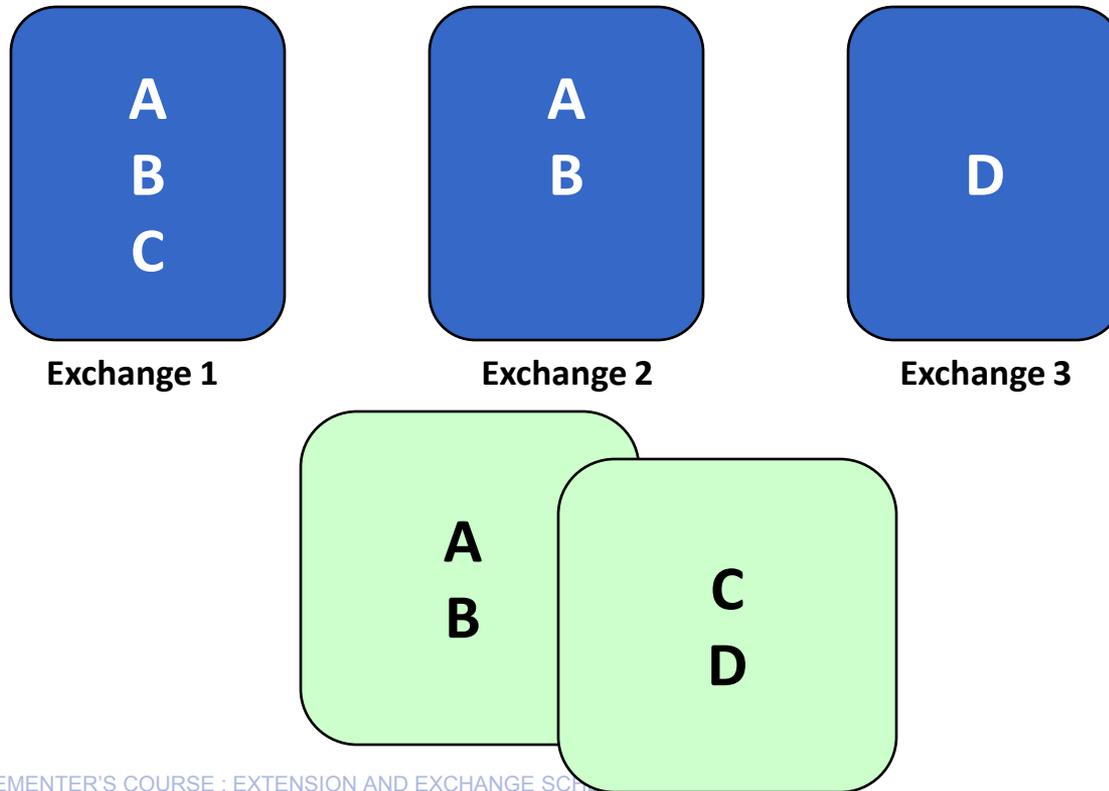
Agency-wide Schema

- Maximizes reuse and consistency within agency.



Multiple Extension Schemas

- May create logical namespaces for extensions



What is an Exchange Schema?

- Defines the root element of an exchange.
- Imports the extension schema, which in turn imports the NIEM schemas.

Creating an Exchange Schema

- Define namespaces
- Import extension schema namespace
- References to NIEM and W3C namespaces
- Define root element

Sample Document Schema Shell

```
<xsd:schema xmlns:root-ns="http://mynamespace.com/exchange/1.0"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:local-ns="http://mynamespace.com/extension/1.0"
xmlns:nc="http://niem.gov/niem/niem-core/2.0" xmlns:i="http://niem.gov/niem/appinfo/2.0"
targetNamespace="http://mynamespace.com/exchange/1.0" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xsd:import namespace="http://mynamespace.com/extension/1.0"
schemaLocation="extension.xsd"/>
    <xsd:element name="MyRootElement" type="local-ns:RootType"/>
</xsd:schema>
```

Recap

- Extension Schemas are included in nearly every IEPD.
- Contain definitions of elements and types not contained in NIEM.
- Are created within their own namespace(s).
- An agency may create extension schemas for each exchange or larger reusable extension schemas.

Issues for Consideration

- Is a separate schema necessary?
 - ◆ Exchange schema makes clear which element is root element.
- Exchange schema requires own namespace.

Case Study

NIEM Practical Implementer's Course

Case Study

- Exchange Schema Development Exercise
 - ◆ Create a basic exchange schema using the files generated from the Case Study exercise in Module 14



Quotation Exchange
Schema Template

Case Study Solution

- Exchange Schema Development Exercise Solution



Simulation Exchange
Schema Solution

Case Study

- Extension Schema Development Exercise
 - ◆ Create the required extension schema



Simulation Extension
Schema Template

Case Study Solution

- Extension Schema Development Exercise Solution



Simulation Extension
Schema Solution

Case Study

- Schema Validation Discussion
 - ◆ Validation of the schema using a parser or schema development tool

Case Study

- Sample XML Exercise
 - ◆ Create sample XML files that validate against the schemas



Citation Sample
XML Template

Case Study Solution

- Sample XML Exercise Solution



Citation Sample
XML Solution

Module Summary

- After completing this module, you should be able to:
 - ◆ Define extension schemas.
 - ◆ Explain the role of extension schemas.
 - ◆ Develop an extension schema.
 - ◆ Differentiate which techniques to use and when creating extension schemas.

Creative Commons



Attribution-ShareAlike 2.0

You are free to

- Copy, distribute, display, and perform the work
- Make derivative works
- Make commercial use of the work

Under the following conditions

- For any reuse or distribution, you must make clear to others the license terms of this work
- Any of these conditions can be waived, if you get permission from the copyright holder

Your fair use and other rights are in no way affected by the above

This is a human-readable summary of the [Legal Code \(the full license\)](#) and [Disclaimer](#)

This page is available in the following languages

[Català](#), [Deutsch](#), [English](#), [Castellano](#), [Suomeksi](#), [français](#), [hrvatski](#), [Italiano](#), [日本語](#), [Nederlands](#), [Português](#), and [中文\(繁\)](#)

[Learn how to distribute your work using this license](#)



Attribution—You must give the original author credit



ShareAlike—If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one